## REMARKS

### Cancelled Claims 1-15 and 17-27

In this Amendment, Applicants have cancelled claims 1-15 and 17-27 from further

5    consideration in this application. Applicant is not conceding that the subject matter

encompassed by claims 1-15 and 17-27, prior to this Amendment is not patentable over the

art cited by the Examiner. Claims 1-15 and 17-27 were cancelled in this Amendment

solely to facilitate expeditious prosecution of the allowable subject matter noted by the

Examiner. Applicant respectfully reserves the right to pursue claims, including the subject

10   matter encompassed by claims 1-15 and 17-27 as presented prior to this Amendment and

additional claims in one or more continuing applications.

### 35 U.S.C. § 103 Claim Rejections

#### Claim 16

By the Office Action dated January 31, 2008, the Examiner has rejected claim 16

15   under 35 U.S.C. § 103(a) as being unpatentable over Fry (U.S. Patent Application

Publication Number 2003/0159111) (hereinafter "Fry") in view of Sikchi (U.S. Patent

Application Publication Number 2004/0193661) (hereinafter "Sikchi"). In order to form a

proper obviousness rejection of a claim under 35 U.S.C. § 103(a), a collection of

references together must teach or suggest each element of the claim, including the

20   relationships between the elements. If any element is not fully taught by the combined

references, the rejection cannot be sustained.

The Examiner asserted that,

Fry discloses in figures 1-2, substantially all of the elements, a

25    client-server system capable of validating cached

eXtensible Markup Language (XML) data comprising

a data store for storing XML data (108, paragraph [0032]);

a server for retrieving and updating XML data in the data store to

service client requests (Abstract, paragraph [0015]);

30    a transformation engine for transforming XML data into a format

suitable for a client application based on a set of transformation

rules (120, 130, paragraph [0049]);

a cache for temporarily storing transformed XML data as data

objects for later reuse (paragraph [0028])

5     .  (See Office Action, pages 4-5.)  The Examiner then admitted that, for claim 16,

Fry discloses in figures 1-2, substantially all of the

elements . . . except

a cache monitor for ensuring that cached objects are validated

10     when changes to XML data in the data store are detected by the

server; and

an object dependency mapper for automatically and

continuously determining dependencies between XML data in

the data store and sets of transformation rules.

15

(See Office Action, page 5.)  The Examiner then asserted that, for claim 16,

Sikchi teaches in figures 1-8, the hierarchical data processing

engine 122 determines, for a node of the data file 126, if

20     possible inputs into that noted require only a simple change to

the rendered form (paragraph [0079]) and this record of the status

of the node can be used by the hierarchical data processing engine

122 or the system 100 to determine how to render a change to this

node.  This record can also include information instructing the

25     hierarchical data processing engine 122 as to which element(s) of a

potential rendering file should be changed for an input to this node

(paragraph [0084]).

(See Office Action, page 5.)  The Examiner finally asserted that, for claim 16,

30     [i]t would have been obvious at the time the invention was made

for one person of the ordinary skill in the art to modify the disclosure

of Sikchi to determine if possible inputs into the node cause only a simple change to data-entry fields when the transformation file is applied on the data file. A simple change, for example, is one in which data rendered in one data-entry field is the data input into the

5        node.

(See Office Action, page 5.)

To the extent the Examiner's language at pages 4 and 5 of the Office Action can be understood, it appears that the Examiner has asserted the following correspondence between Fry and Sikchi and claim 16:

| Claim 16 | Fry | Sikchi |
|---|---|---|
| 16.      In a client-sever computing system having a cache and storing *eXtensible Markup Language (XML)* data as data objects, a method for determining invalid cached objects comprising: | - | Sikchi does not teach this claim element. |
| transforming *XML* data into a format suitable for a client application based on a set of transformation rules; | Fry does not teach this claim element. | Sikchi does not teach this claim element. |
| determining dependencies between cached objects and *XML* data related to the cached objects; | Fry does not teach this claim element. | Sikchi does not teach this claim element. |
| monitoring updates to the related *XML* data; and | Fry does not teach this claim element. | Sikchi does not teach this claim element. |

| determining the cached objects that are affected by changes to the related *XML* data based on the dependencies. | Fry does not teach this claim element. | Sikchi does not teach this claim element. |
|---|---|---|

In reviewing the cited portions of Fry and Sikchi, however, it becomes apparent that Fry and Sikchi have been generalized, and, in fact, does not support the position asserted by the Examiner.

**transforming *XML* data into a format suitable for a client application based on a set of transformation rules**

In particular, Fry and Sikchi, alone or in combination, fail to teach or suggest "transforming XML data into a format suitable for a client application based on a set of transformation rules", as required by claim 16.  The Examiner cites items "120, 130, paragraph [0049]" in Fry in asserting that Fry teaches "a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation rules".  However, Fry does not include items 120, 130 in its drawings or text and does not include a paragraph [0049].  Thus, Fry cannot teach or suggest "a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation rules". Thus, Fry fails to teach or suggest "transforming XML data into a format suitable for a client application based on a set of transformation rules".  The Examiner did not assert that Sikchi teaches or suggests "a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation rules".  Thus, Sikchi cannot teach or suggest "a transformation engine for transforming XML data into a format suitable for a client application based on a set of transformation rules". Thus, Sikchi fails to teach or suggest "transforming XML data into a format suitable for a client application based on a set of transformation rules".  Therefore, Fry and Sikchi, alone or in combination, cannot teach or suggest the claim 16 element of "transforming XML data into a format suitable for a client application based on a set of transformation rules".

**determining dependencies between cached objects and *XML* data related to the cached objects**

In particular, <u>Fry</u> and <u>Sikchi</u>, alone or in combination, fail to teach or suggest "determining dependencies between cached objects and *XML* data related to the cached

5    objects", as required by claim 16.  The Examiner admitted that <u>Fry</u> fails to teach or suggest "an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules".  (See Office Action, page 5.)  Thus, the Examiner admitted that <u>Fry</u> fails to teach or suggest "determining dependencies between cached objects and *XML* data related to the cached

10   objects".  <u>Sikchi</u> discusses

> At block 402, the system 100, through the hierarchical data
> Processing engine 122, analyzes the transformation file 128 for
> isolatable nodes and subtrees. An isolatable node is one in which
> data input into that node affects the rendering file 130 only by

15   > changing its corresponding data-entry field(s) by replacing the data
> within that data-entry field(s) with the data input. An isolatable
> subtree of nodes depends only on nodes within the subtree.

(See <u>Sikchi</u>, paragraph [0046].)  <u>Sikchi</u> does not teach "objects" or "determining dependencies between cached objects and *XML* data".  Thus, <u>Sikchi</u> fails to teach or

20   suggest teach or suggest "an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation rules".  Thus, <u>Sikchi</u> fails to teach or suggest "determining dependencies between cached objects and *XML* data related to the cached objects".  Therefore, <u>Fry</u> and <u>Sikchi</u>, alone or in combination, cannot teach or suggest the claim 16 element of "determining dependencies

25   between cached objects and *XML* data related to the cached objects".

**monitoring updates to the related *XML* data**

Also, <u>Fry</u> and <u>Sikchi</u>, alone or in combination, fail to teach or suggest "monitoring updates to the related *XML* data", as required by claim 16.  The Examiner admitted that <u>Fry</u> fails to teach or suggest "a cache monitor for ensuring that cached objects are validated

30   when changes to *XML* data in the data store are detected by the server".  (See Office Action, page 5.)  Thus, the Examiner admitted that <u>Fry</u> fails to teach or suggest

"monitoring updates to the related *XML* data". Sikchi does not disclose a cache. Thus, Sikchi fails to teach or suggest "a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server". Thus, Sikchi fails to teach or suggest that cache manager can be used for "monitoring updates to

5    the related *XML* data". Therefore, Sikchi cannot teach or suggest "monitoring updates to the related *XML* data". Therefore, Fry and Sikchi, alone or in combination, cannot teach or suggest the claim 16 element of "monitoring updates to the related *XML* data".

**determining the cached objects that are affected by changes to the**
**related *XML* data based on the dependencies**

10   In addition, Fry and Sikchi, alone or in combination, fail to teach or suggest "determining the cached objects that are affected by changes to the related *XML* data based on the dependencies", as required by claim 16. The Examiner admitted that Fry fails to teach or suggest "an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation

15   rules". (See Office Action, page 5.) The Examiner also admitted that Fry fails to teach or suggest "a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server". (See Office Action, page 5.) Thus, the Examiner admitted that Fry fails to teach or suggest" determining the cached objects that are affected by changes to the related *XML* data based on the dependencies". Sikchi

20   discusses

> At block 402, the system 100, through the hierarchical data
> processing engine 122, analyzes the transformation file 128 for
> isolatable nodes and subtrees. An isolatable node is one in which data
> input into that node affects the rendering file 130 only by

25   > changing its corresponding data-entry field(s) by replacing the data
> within that data-entry field(s) with the data input. An isolatable
> subtree of nodes depends only on nodes within the subtree.

(See Sikchi, paragraph [0046].) Sikchi does not teach "objects" or "determining dependencies between cached objects and *XML* data". Thus, Sikchi fails to teach or

30   suggest teach or suggest "an object dependency mapper for automatically and continuously determining dependencies between *XML* data in the data store and sets of transformation

rules". <u>Sikchi</u> does not disclose a "cache". Thus, <u>Sikchi</u> fails to teach or suggest "a cache monitor for ensuring that cached objects are validated when changes to *XML* data in the data store are detected by the server". Thus, <u>Sikchi</u> fails to teach or suggest either an object manager or a cache manager that can be used for "determining the cached objects

5    that are affected by changes to the related *XML* data based on the dependencies". Therefore, <u>Sikchi</u> cannot teach or suggest "determining the cached objects that are affected by changes to the related *XML* data based on the dependencies". Therefore, <u>Fry</u> and <u>Sikchi</u>, alone or in combination, cannot teach or suggest the claim 16 element of determining the cached objects that are affected by changes to the related *XML* data based

10   on the dependencies". It is therefore clear that <u>Fry</u> and <u>Sikchi</u>, alone or in combination, cannot teach or suggest each element of claim 16 and, therefore, a rejection of claim 16 under 35 U.S.C. § 103(a) would be inappropriate.

**Conclusion**

It is therefore clear that claim 16 complies with the requirements of 35 U.S.C. §§

15   101, 102, 103, and 112. The application is therefore in condition for allowance. Early notification to that effect is respectfully solicited.

In the event that any issue remains unresolved, the Examiner is invited to telephone the undersigned at 408-927-3377.


20                                                          Respectfully Submitted,


                                                           /Leonard T. Guzman/
     Date:  June 11, 2008                                  Leonard T. Guzman
                                                           Reg. No. 46,308

25

     IBM Almaden Research Center
     650 Harry Road
     C45A/J2B
     San Jose, CA 95120
30   Phone Number:        408-927-3377
     Facsimile Number:    408-927-3375